



곤돌라

마오공 곤돌라는 타이페이의 명소 중 하나이다. 곤돌라 시스템은 원형의 레일과, 하나의 정류장이 있고, 1 부터 n 까지 순서대로 번호가 붙은 n 개의 곤돌라가 모두 단일한 방향으로 레일을 따라 움직이는 형태이다. i 번 곤돌라가 정류장을 지난 직후에는 $i + 1$ 번 곤돌라가 정류장을 지나가게 된다. (단, $i = n$ 번 곤돌라가 지나간 직후에 1 번 곤돌라가 지나가게 된다.)

곤돌라들은 고장이 나기도 한다. 다행히 무제한으로 많은 곤돌라의 여분이 있고, 여분 곤돌라들은 $n + 1, n + 2$ 과 같이 순차적으로 번호가 붙어 있다. 특정한 곤돌라가 고장이 나면 고장난 곤돌라는 빼고, 동일한 위치에 여분 곤돌라를 배치한다. 여분 곤돌라는 작은 번호부터 사용된다. 예를 들어, 사용하는 곤돌라 수가 총 5 개이고, 1 번 곤돌라가 고장난다면, 그 곤돌라는 6번으로 교체된다.

당신은 정류장에 서서 곤돌라들이 지나가는 것을 즐겨 본다. **곤돌라 수열**이라는 것은 임의의 시점에서 시작해서 정류장을 지나가는 n 개의 곤돌라들의 번호를 순서대로 적은 것이다. 곤돌라 수열을 적기 시작하는 시점 이전에 이미 몇개의 곤돌라가 고장나서 교체되었을 수 있다. 하지만, 곤돌라 수열을 적는 도중에는 아무 곤돌라도 고장이 나지 않는다.

전체적으로 곤돌라들의 배치가 동일하더라도 어떤 시점에 곤돌라 수열을 적기 시작하느냐에 따라서 다른 곤돌라 수열이 나올수 있다는 점에 주의하자. 예를 들어, 총 5개의 곤돌라들 중 고장난 곤돌라가 없는 경우에 (2, 3, 4, 5, 1) 과 (4, 5, 1, 2, 3) 은 모두 가능한 곤돌라 수열들이다. 하지만, 이 경우 (4, 3, 2, 5, 1) 은 가능한 곤돌라 수열이 아니다. (곤돌라 번호의 순서가 잘못되어 있다.)

만약 곤돌라 1번 만이 고장난 상황이라면, (4, 5, 6, 2, 3) 의 곤돌라 수열을 만들 수 있다. 만약 이후 4번 곤돌라가 고장난다면, 7번 곤돌라가 그 자리에 있게 되고, (6, 2, 3, 7, 5) 가 가능한 곤돌라 수열이 된다. 만약 7번 곤돌라가 이후에 고장이 난다면, 8번이 그 자리를 차지할 것이고 (3, 8, 5, 6, 2) 가 가능한 곤돌라 수열들 중 하나가 된다.

고장난 곤돌라	새 곤돌라	가능한 곤돌라 수열 중 하나
1	6	(4, 5, 6, 2, 3)
4	7	(6, 2, 3, 7, 5)
7	8	(3, 8, 5, 6, 2)

교체 수열이라는 것은 고장난 곤돌라들의 번호를 고장난 순서에 따라 쓴 것이다. 직전의 예에서 교체 수열은 (1, 4, 7) 이다. 교체 수열 r 이 곤돌라 수열 g 를 **만든다**고 말할 수 있는데, 그것은 초기 상황에서 시작해서 r 에 해당하는 방법으로 곤돌라들이 고장난 직후에, g 가 가능한 곤돌라 수열들 중 하나인 경우를 의미한다.

곤돌라 수열 확인

처음 세 개의 부분 문제에서는 입력 수열이 곤돌라 수열로서 가능한 것인지 확인하여야 한다. 몇 개의 곤돌라가 고장나서 교체된 상태일 수도 있다. 아래 표에는 곤돌라 수열인 예들과 곤돌라 수열이 아닌 예들이 나와 있다. 아래와 같이 선언된 valid 함수를 구현하여야 한다.

- `valid(n, inputSeq)`
 - `n`: 입력 수열의 길이.
 - `inputSeq`: 크기 `n` 인 배열; `inputSeq[i]` 는 입력의 `i` 번 원소이다 ($0 \leq i \leq n - 1$).
 - 함수는 입력이 곤돌라 수열로서 가능한 경우 1을, 아닌 경우 0을 리턴해야 한다.

부분 문제 1, 2, 3

부분 문제	점수	n	<code>inputSeq</code>
1	5	$n \leq 100$	1 부터 n 까지의 수들이 정확히 한번씩 나온다
2	5	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq n$
3	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$

예제

부분 문제	<code>inputSeq</code>	리턴 값	비고
1	(1, 2, 3, 4, 5, 6, 7)	1	
1	(3, 4, 5, 6, 1, 2)	1	
1	(1, 5, 3, 4, 2, 7, 6)	0	1 이 5 의 직전에 나오는 것은 불가능
1	(4, 3, 2, 1)	0	4 가 3 의 직전에 나오는 것은 불가능
2	(1, 2, 3, 4, 5, 6, 5)	0	5 번 곤돌라가 두개
3	(2, 3, 4, 9, 6, 7, 1)	1	교체 수열 (5, 8) 로 가능
3	(10, 4, 3, 11, 12)	0	4 가 3 의 직전에 나오는 것은 불가능

교체 수열

다음 세 개의 부분 문제들에서는 주어진 곤돌라 수열을 만들 수 있는 교체 수열을 생성하여야 한다. 여러 개의 교체 수열이 가능한 경우 그 중 하나를 생성하면 된다. 다음과 같이 선언된 함수 `replacement` 를 구현해야 한다.

- `replacement(n, gondolaSeq, replacementSeq)`
 - `n`: 입력 곤돌라 수열의 길이이다.
 - `gondolaSeq`: 크기 `n` 인 배열; `gondolaSeq` 는 항상 가능한 곤돌라 수열이며, `gondolaSeq[i]` 는 `i` 번 원소이다 ($0 \leq i \leq n - 1$).
 - 함수는 교체 수열의 길이 `l` 을 리턴해야 한다.
 - `replacementSeq`: 교체 수열을 저장하기에 충분한 크기의 배열; `replacementSeq[i]` 에는 계산된 교체 수열의 `i` 번 원소가 저장되어야 한다 ($0 \leq i \leq l - 1$).

부분 문제 4, 5, 6

부분 문제	점수	n	<code>gondolaSeq</code>
4	5	$n \leq 100$	$1 \leq \text{gondolaSeq}[i] \leq n + 1$
5	10	$n \leq 1,000$	$1 \leq \text{gondolaSeq}[i] \leq 5,000$
6	20	$n \leq 100,000$	$1 \leq \text{gondolaSeq}[i] \leq 250,000$

예제

부분 문제	<code>gondolaSeq</code>	리턴 값	<code>replacementSeq</code>
4	(3, 1, 4)	1	(2)
4	(5, 1, 2, 3, 4)	0	()
5	(2, 3, 4, 9, 6, 7, 1)	2	(5, 8)

교체 수열의 개수 세기

다음 네 개의 부분 문제들에서는 주어진 수열(가능한 곤돌라 수열일 수도 아닐 수도 있음)을 만들 수 있는 교체 수열의 수를 세어서 그 값을 **1,000,000,009** 로 나눈 나머지를 계산해야 한다. 다음과 같이 선언된 함수 `countReplacement` 를 구현해야 한다.

- `countReplacement(n, inputSeq)`
 - n : 입력 수열의 길이.
 - `inputSeq`: 크기 n 인 배열; `inputSeq[i]`는 입력 수열의 i 번 원소이다 ($0 \leq i \leq n - 1$).
 - 입력이 곤돌라 수열인 경우, 입력의 곤돌라 수열을 만들 수 있는 모든 교체 수열의 수를 센 다음에 (그 수는 매우 클수 있음) 그 결과를 **1,000,000,009** 로 나눈 나머지를 리턴해야 한다. 만약 입력이 곤돌라 수열이 아닌 경우 0을 리턴해야 한다. 입력이 곤돌라 수열이고 고장난 곤돌라가 하나도 없는 경우는 1을 리턴해야 한다.

부분 문제 7, 8, 9, 10

부분 문제	점수	n	<code>inputSeq</code>
7	5	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq n + 3$
8	15	$4 \leq n \leq 50$	$1 \leq \text{inputSeq}[i] \leq 100$ 이고 초기 곤돌라들 중 (즉, $1, \dots, n$ 번 중) 최소한 $n - 3$ 개는 고장이 나지 않았다.
9	15	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 250,000$
10	10	$n \leq 100,000$	$1 \leq \text{inputSeq}[i] \leq 1,000,000,000$

예제

부분 문제	inputSeq	리턴 값	교체 수열
7	(1, 2, 7, 6)	2	(3, 4, 5) or (4, 5, 3)
8	(2, 3, 4, 12, 6, 7, 1)	1	(5, 8, 9, 10, 11)
9	(4, 7, 4, 7)	0	inputSeq가 곤돌라 수열이 아님
10	(3, 4)	2	(1, 2) or (2, 1)

구현 주의 사항

단 하나의 파일을 제출해야 한다. 이름은 `gondola.c`, `gondola.cpp`, 혹은 `gondola.pas` 이다. 이 파일에는 위에서 말한 세가지 함수가 다 존재해야 한다. (부분 문제들 중 일부만 풀려고 하는 경우도 마찬가지이다.) 다음의 함수 선언을 이용해야 한다. C/C++ 구현에서는 `gondola.h` 를 `#include`해야 한다.

C/C++ programs

```
int valid(int n, int inputSeq[]);
int replacement(int n, int gondolaSeq[], int replacementSeq[]);
int countReplacement(int n, int inputSeq[]);
```

Pascal programs

```
function valid(n: longint; inputSeq: array of longint): integer;
function replacement(n: longint; gondolaSeq: array of longint;
var replacementSeq: array of longint): longint;
function countReplacement(n: longint; inputSeq: array of longint):
longint;
```

Sample grader

Sample grader의 입력 양식은 다음과 같다.

- line 1: T , 풀려고 하는 부분 문제의 번호 ($1 \leq T \leq 10$).
- line 2: n , 입력 수열의 길이.
- line 3: T 가 4, 5, 6 인 경우, 이 줄에는 `gondolaSeq[0], ..., gondolaSeq[n-1]` 가 있고, 그렇지 않은 경우 `inputSeq[0], ..., inputSeq[n-1]` 가 있다.